

## REMARKS

Reconsideration of the above-identified application in view of the amendments above and the remarks following is respectfully requested.

The Examiner objected to Figures 1a and 1c which illustrate prior art and were not designated as such. Replacement sheets are included herewith.

Claims 1-17 were rejected under 35 U.S.C § 112, second paragraph and under 35 U.S.C § 101.

Claims 1-8 and 11-16 are rejected under 35 U.S.C § 102(a) as being anticipated by Lie et al., "Architectural Support for Copy and Tamper Resistant Software".

Claims 1, 12 and 17 are rejected under 35 U.S.C § 102(b) as being anticipated by McManis EP 0770957

Claims 9,10 and 17 are rejected under 35 U.S.C § 103 as being unpatentable over Lie et al. in view of Saito et al. US patent 6,081,794.

While continuing to traverse the Examiner's rejections, and without in any way prejudicing the patentability of the rejected claims, the Applicant has, in order to expedite the prosecution, chosen to amend the claims thereby rendering moot Examiner's rejections.

Applicant has rewritten all claims more particularly and distinctly so as to overcome the technical rejections and define the invention patentably over the prior art. Claims 1-17 are hereby canceled without prejudice and are replaced by new claims 18-37.

### **The References and Differences of the Present Invention Thereover:**

Prior to discussing the claims, Applicant will first discuss the references of the prior art of record and the novelty of the present invention and its unobviousness over the references.

By way of introduction, Applicant respectfully affirms that there are fundamental differences between the Lie et al., "Architectural Support for Copy and Tamper Resistant Software" and the present invention. The disclosed architecture of Lie et al. is based on embedding hardware modifications to a computer processor. The modifications then enable the modified processor to execute the copy and tamper

resistant software that was encrypted specifically for the modified processor. Thus, executing protected software according to Lie et al. requires software users to have a computer system with a modified processor assuming the current processor supports XOM code. Most commercial processors do not support XOM architecture.

The present invention on the other hand, relates to a method for authenticating and protecting digital data from illegal copy and use, which is independent of the processor or any other hardware. Thus, protected software according to the present invention can be executed on any computer hardware. Moreover, software vendors that wish to protect their software, according to Lie et al., must encrypt each copy of the software/content for the specific processor of each user. All the recipients of the software must be known (or at least their processors) and each copy of the software must be delivered to a specific recipient. A software copy delivered to a specific user will immediately become useless when the user replaces/upgrades his processor or replaces his entire computer system. The rapid enhancements nowadays in computer systems and processors especially, cause software consumers to upgrade their computer systems very frequently, rendering impractical the protection of software according Lie et al. Furthermore, many software/content consumers own more than one computer system. The protected software according to Lie et al. can be executed/decrypted only by a specific processor, rendering the disclosure of Lie et al. impractical for software vendors. Some software is designed to work on multiple computers (clusters of computers). For example, to increase performance by balancing the load of processing tasks on several computers. Sometimes each computer may have a dedicated task such as sound processing or video processing, and in other cases, the same code is required to run on several different computers. Since the disclosure of Lie requires that the software code will be encoded specifically for each processor this task becomes too complex and impractical.

According to the present invention, software vendors that wish to protect their software according to the present invention, are not limited in their distribution methods. For example, unlimited number of copies of the software can be produced and distributed via retail stores regardless of the computer in use by the end user. The receiver of the protected software or content may execute the software or present the content on any computer system as long as the software/content is stored on the original medium. Lie et al. does not include or suggest any of the elements of the present invention as amended herein.

McManis EP 0770957 discloses a computer system that has a program module verifier and multiple program modules. Each program module includes a digital signature and an executable procedure. A first program module furthermore includes a procedure call to a second procedure module. The procedure call to the program module verifier that is logically positioned in the first program module so as to be executed prior to execution of the procedure call to the second program module. Execution is prevented of the second program module when the procedure call to the program module verifier results in a verification denial being returned by the program module verifier.

Applicant was surprised by the 102(b) rejection based on McManis. McManis is directed towards verification of executable modules in random access memory during execution. McManis does not describe a method for protecting storage media from being copied and as such McManis does not disclose or even suggest any of the elements of the present invention as amended herein. Applicant is submitting herewith a favorable opinion from an International Preliminary Examination Report of a parallel application PCT/IL02/00143 with claims similar to canceled claims 1-17 of the present application which cites McManis as the nearest prior art.

Saito et al., US patent 6081794 discloses a data copyright management system in which a primary user edits received data and supplies the edited data to a secondary user. The copyright management system includes a database and a key control center, and uses a primary copyright label, a primary use permit key including a first crypt key, a secondary use permit key, a third crypt key, and a copyright management program. The primary user decrypts the copyrighted primary data, which is encrypted using the first crypt key and supplied, to plaintext using a primary use permit key obtained from the key control center. If the copyrighted primary data is stored in a primary user device, it is re-encrypted using the primary use permit key. The primary user receives a secondary use permit key for editing the copyrighted primary data from the key control center and edits the copyrighted primary data. The data being edited is encrypted using the secondary use permit key and is stored. When edit has been completed, the primary user receives a third crypt key for secondary use with a secondary copyright from the key control center, and encrypts the edited data using the third crypt key and distributes it to the secondary user. The secondary user receives the third crypt key from the key control center and utilizes the edited data. The third crypt key may be generated by the primary user or by the key control center.

The method as disclosed by Saito et al. is a rather esoteric example of using software on a "pay per" basis amply described in the background of the present invention on page 4 paragraph. 4. Saito et al. does not disclose nor suggest a method for securing and preventing unauthorized copying of media, according to the methods of the present invention as amended herein.

Similarly, Applicant has carefully reviewed all prior art of record and found none which are relevant at all to the present invention.

Applicant respectfully points out that claims rejections based on 35 U.S.C § 102(a) (page 10 from line 27) are not appropriate since the prior art references on record have very different purposes and are therefore not mutually compatible and hence inoperable if the references are combined.

**Independent Claim 1 now written as claim 18 and Claim 37**

Applicant has rewritten Claim 1 as new Claim 18 in order to overcome technical rejections and clarify patentability over the prior art. Applicant respectfully traverses Examiner's interpretation (page 5 lines 21-22 of the present office communication) that "encrypted blocks" as mentioned in Lie et al. are equivalent to "mines" or software procedures of the present invention. Lie et al. is describing and using a method well known in the prior art as cipher block chaining in which each block of plaintext is XORed with the previous ciphertext block before being encrypted. This way, each ciphertext block is dependent on all plaintext blocks up to that point. The term "mines" of the present invention, (as defined on page 8 of the present application) is a software procedure or executable code which performs a security function such as checking the authenticity and/or validity of data in use. The additional functionality of a mine differentiates the present invention from Lie et al. which is based on encrypted blocks without the additional functionality. Proper operation/use of the executable program file depends on one or more keys: validation key , authentication key and/or signature key which enable the additional functionality (*e.g.* authentication and validation of the data). In Lie et al. the block chaining is designed to prevent relocation of code segments by an adversary.

Applicant has in claim 18, replaced the term "mine" to "software procedure". which performs the security function. Claim 18 includes limitations of *designating* locations within an executable program file, *arming* the program file by inserting the "software procedures" or mines, and *storing* the modified executable file and

*executing* the software procedure which performs the security function (in addition to being encrypted/decrypted). Lie et al. (nor any other prior art references of record) do not disclose nor suggest any of the steps of claim 18. Regarding the limitation "*executing*", the guarding module of the prior art, in contrast runs constantly in the background while a "mine" or "software procedure" of the present invention does not run in the background and is activated when the mine is reached during execution of the protected software or content.

Applicant wishes to point out "new and unexpected results" of using distinct steps *designating* and *arming*. *Designating* is used just to mark the software parts to be protected and generate a flagged software file. The flagged software doesn't include any additional functionality comparing to the original un-flagged software. At a later stage, during the *arming* step, a machine code containing security functionality is inserted into the marked locations at the flagged software. The use of two distinct steps *designating and arming* is essential for commercial utilization, because the provider of the copy protection (*e.g.* a CD replication facility) doesn't wish to grant to the software vendor, author or copyright owner fully functional tools that can be used for an unlimited number of copies. Instead, the tools delivered to the software vendor do not add any functionality, but just mark the locations in the software to be protected. The copy protection provider can then perform the arming process on the number of copies requested by the software vendor (*e.g.* replicating 1000 CDs containing the armed software) and charge the software vendor for the copy protection service according to the number of copies on which the arming process was performed. This process is illustrated in figures 1d and 1e.

### **Dependent Claims**

Although Applicant submits that independent claim 18 includes novel steps and is not obvious, thereby rendering all dependent claims therefrom also patentable, Applicant wishes to briefly point out at least some of the patentable aspects of dependent claims in their own right.

Regarding new dependent claim 20, the dependency among mines in the present invention is different from the "block chaining encryption technique" disclosed in Lie because in the present invention the "mines" are self-decrypting, whereas in Lie et al. decryption of chained blocks is dependent on external

hardware. In Lie et al., for instance, the decryption of a chained block is performed by the processor using previously fetched instructions. In the present invention, each mine themselves "fetches" the necessary information from another mine using a relative address hard coded into the instructions of the fetching mine. Self-decryption is essential for rendering versatility to the present invention, as previously discussed for instance for changing users or upgrading computers.

Applicant respectfully traverses Examiner's rejection of claim 2, (page 6 line 6 of present application). Claim 2 is rewritten as new claim 26. In the present invention, signature keys and content keys are stored together with the protected data on the same medium. Therefore the owner of the specific original medium can transport the medium to any other computer system and use the protected data based on his sole decision without requiring permission such as a certificate from "a trusted certificate authority" *e.g.* VeriSign. The present invention can also be utilized using a user authentication key for even removing the dependency of the specific medium. These properties make the present invention much more flexible and versatile than Lie and other prior art which do not store signature and content keys on the same medium.

Applicant respectively traverses Examiner's rejection (page 8 line 2 of present communication) of claim 11, now rewritten as claim 23. Updating a private key of a processor as disclosed in Lie et al. is very different from the use of authentication keys accessible via the Internet according to the new invention, in both functionality and purpose. According to Lie et al., the purpose of replacing a private key replaces the identity of the processor and does not enable specific software to operate on the specific processor, since replacing the private key will also disable the operation of all other software using the current private key. Enabling software to run on a specific processor is achieved only by encrypting the software with the processor's private key and delivering the entire new copy of the software to the user. According to the present invention, one of the purposes of the authentication key is to allow the operation of the protected software. Furthermore, different software vendors, can allow a specific user to run their software at the same time on a single processor, where each software is protected according to the present invention utilizing a different authentication key (accessible via the Internet). This cannot be achieved by the disclosure of Lie et al.

**Commercial Success as Secondary Consideration showing  
New and Unexpected Results**

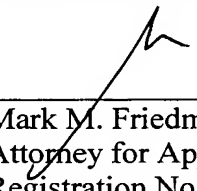
Applicant wishes to point out that the present invention, for the reasons stated previously, has been successfully utilized on millions of copies of commercial software and other digital content published by major publishers around the world as well as government and military organizations. The present invention is utilized successfully in protecting all kinds of digital information carried on various media types such as CD-ROM, recordable discs and the Internet. The present invention is commercially available worldwide via HexaLock Ltd (<http://www.hexalock.com>).

-----

The new claims are fully supported by the specification and new matter has not been added in the present amendment.

In view of the above amendments and remarks it is respectfully submitted that independent claims 18, and 37 and claims dependent therefrom are in condition for allowance. Prompt notice of allowance is respectfully and earnestly solicited.

Respectfully submitted,

---

Mark M. Friedman  
Attorney for Applicant  
Registration No. 33,883

Date: Nov 27, 2005